HMM-based Online Handwriting Recognition System for Telugu Symbols

V. Jagadeesh Babu, L. Prasanth, R. Raghunath Sharma, G.V. Prabhakara Rao,
Sri Sathya Sai Institute of Higher Learning
Puttaparthy, India

Bharath A
Hewlett-Packard Labs
Bangalore, India

Abstract

In this paper we present an online handwritten symbol recognition system for Telugu, a widely spoken language in India. The system is based on Hidden Markov Models (HMM) and uses a combination of time-domain and frequency-domain features. The system gives top-1 accuracy of 91.6% and top-5 accuracy of 98.7% on a dataset containing 29,158 train samples and 9,235 test samples. We also introduce a cost-effective and natural data collection procedure based on ACECAD® Digimemo® and describe its usage in building a Telugu handwriting dataset.

1. Introduction

The need for better man-machine interface is increasing with the development of technology. Due to the large alphabet size in the case of Indic scripts, interaction with the computer using the conventional keyboard has long been a bottleneck. One has to remember the combination of keys to input a character. However, with the development of pen-based devices such as Tablet PC and PDA, handwritten input for text entry provides a more natural alternative which not only solves the problem of large alphabet size but also helps in extending the reach of Information Technology to a larger community. Hence, handwriting recognition for text input acquires great significance in the context of Indic scripts.

In an online handwriting recognition system, the input is a temporal sequence of X-Y points representing the pen trajectory, captured using a digitizer that senses the pen-tip position while writing. While substantial amount of research work has been carried out on handwriting recognition for Western and CJK (Chinese, Japanese and Korean) scripts, Indic script recognition has received relatively less attention. In this paper, we present an online handwriting recognition system to recognize the symbols in Telugu.

Telugu is one of the official languages of India. It is a Dravidian language and has several million speakers across the globe. There have been very few research attempts in the literature, targeted at online Telugu script recognition. The efforts published in [1] and [2] are based on stroke-level recognition. In [1], each stroke is represented as a string of features and then compared with a database of strokes for recognition. In the work described in [2], Support Vector Machines (SVM) and Hidden Markov models (HMM) are used to model strokes. Once the input strokes are identified, a rule-based system recognizes the input character.

In this paper, we describe a symbol recognition system for Telugu, based on continuous density Hidden Markov Models. The recognition system contains four stages: data collection, preprocessing, feature extraction and recognition. These stages are described in Sections 2, 3, 4 and 5 respectively. Results obtained in our experiment are presented in Section 6. In Section 7, conclusions and our future work are discussed.

2. Data Collection

Like most Indic scripts, Telugu is derived from the Brahmi script and is from the family of syllabic alphabets [3]. There are 18 vowels and 36 consonants, of which 13 vowels and 35 consonants are in common usage. A syllabic unit could be a vowel (V), consonant (C) or one of their combinations. The combinations include CV, CC, CCV and also CCCV. In a CV combination, the vowel part is indicated using a diacritic sign known as *maatra*. The shape of a *maatra* is often completely different from the corresponding vowel. The shape of a consonant also changes when it combines with a vowel or with another consonant. These complex composition rules and a large number of syllabic units in the script make the recognition task harder when compared to Western scripts and some of the other Indic scripts like Tamil.

Theoretically, the number of syllabic units is of the order of thousands but a much smaller subset is used in practice. Nevertheless, considering each syllabic unit as a distinct class would increase the complexity of recognition. The approach adopted here is to identify a smaller subset of basic units that are sufficient to cover the entire set of syllabic units in the script. These basic

units are defined taking into account the burden involved in data collection and recognition. For example, certain vowel *maatras* in Telugu appear inseparable from the base consonant as in the case of 'చి'. In such a case, considering the consonant and the *maatra* as separate recognition units would require the syllabic unit to be segmented before recognition. Therefore in our experiment, the basic units are determined considering various factors such as ease of segmentation of syllabic units, possibility of sharing stable shapes across syllabic units etc., and not just based on linguistic criteria [4].

2.1. Telugu Symbol Set

The basic graphemes of the script i.e. independent vowels, consonants, half-consonants and *maatras* are included in our symbol set defined for recognition. Some consonant-vowel units which cannot be easily segmented, and symbols which do not have linguistic interpretation but have stable shapes across syllabic units have also been added to the set to reduce the effort involved in data collection.

The complete symbol set containing a total of 141 symbols that cover the entire Telugu script is shown in Fig.1.

Figure 1. Symbol set for Telugu

2.2. Digimemo-based Data Collection Tool

Even though interactive devices such as the Tablet PC and PDA appear to be appropriate for online handwriting data collection, they fail to recreate one's feel of writing on paper. Native writers are typically unfamiliar with such devices. In order to address these issues, we used the ACECAD Digimemo [5] for data collection. The Digimemo is a portable device which digitally captures and stores the ink written on ordinary paper using a digital pen and pad. Hence it provides a natural interface for collecting handwriting data samples. It also provides an affordable alternative to devices such as TabletPC and PDAs for larger scale data collection efforts.

The writers who participated in the data collection activity were provided with A5-sized booklets clipped to the Digimemo pad. Each sheet contained the symbols to be written and empty boxes for writing them. A sample filled-in page is shown in Fig.2. The ink captured by the device from each of the boxes was extracted and stored in UNIPEN [6] format. The output of data collection is a set of UNIPEN files containing the digital ink, and meta-data about the writer profile and collection procedure.

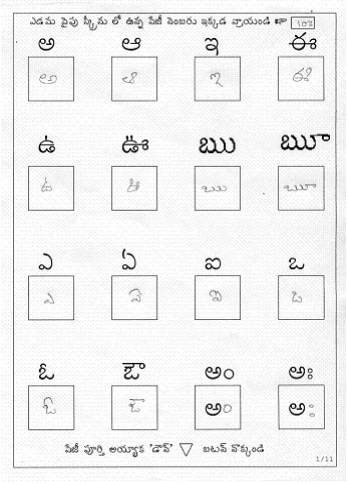

Figure 2. A sample data collection form

The ACECAD Digimemo is easy to use, portable, cost-effective and most importantly provides the writer with the natural feel of writing on paper. However, the location of the sensor above the pen tip poses a peculiar problem. The coordinates recorded by the digitizer are offset by an amount proportional to the degree of pen tilt while writing. The degree of tilt not only varies with the writer but also depends on which part of the page the writer is writing on. The amount of tilt needs to be determined both to identify the strokes that are falling inside a box and to preserve the relative position of the symbol with respect to the writing box. The relative position information becomes essential if the collected symbol-level data were to be used for recognition of complete syllabic units. To determine the tilt for each writer and for each box, the writer is first provided with a "tilt calibration" form containing the printed boxes similar to the pages that contain symbols, except for the presence of a dotted 'X' mark in each of the boxes. The writer is asked to trace the

dotted 'X' in each of the boxes and the difference between the intersection point of the cross mark and the actual centre of the box is taken as the offset caused due to pen tilt. For each writer, the offset is found for every box of the page. This method of finding the offset assumes that the writer writes on subsequent pages with approximately the same degree of tilt as he does on the tilt calibration page. Although the offset compensation method is an approximate one, it worked well in practice.

A total of 38,393 data samples were collected from 143 writers belonging to different age groups, genders and educational backgrounds.

3. Preprocessing

The preprocessing stage of the system involves duplicate point elimination, smoothing, size normalization and resampling.

Duplicate points are those that have identical X-Y values as that of the preceding point and do not contain any information for recognition. Hence they are removed before feature extraction. Smoothing is required to remove any noise in the writing due to erratic pen motion. A moving average window of fixed size is used for smoothing. Size normalization is necessary to remove variations due to size of the writing. It is achieved by fitting the sample into a square of length 10. In digital ink capture, the points appear equidistant in time but not in space. Hence, the number of points varies depending on the speed of writing and the sampling rate of the digitizer. In order to remove these variations, the coordinate sequence is resampled spatially along the trajectory. Each symbol sample is resampled to 30 points by linear interpolation. In the case of multi-stroke symbols, the points are allocated to each stroke based on its relative stroke length.

4. Feature Extraction

In this stage, each preprocessed sample is transformed into a sequence of feature vectors. At each point in the sample, a 19-dimensional feature vector which consists of 11 time-domain features and 8 frequency domain features is computed.

4.1. Time-domain features

The time-domain features are largely adapted from [7, 8] and are described below.

- **Normalized x-y coordinates**: The x and y coordinates from the normalized sample constitute the first 2 features.
- **Normalized first derivatives**: The normalized first derivatives $\hat{x}'$ and $\hat{y}'$ are calculated as in [7].

$$x'_i = \frac{\sum_{i=1}^{2} i.(x_{i+1} - x_{i-1})}{2.\sum_{i=1}^{2} i^2} \qquad y'_i = \frac{\sum_{i=1}^{2} i.(y_{i+1} - y_{i-1})}{2.\sum_{i=1}^{2} i^2}$$

$$\hat{x}'_i = \frac{x'_i}{\sqrt{x'_i{}^2 + y'_i{}^2}} \quad ; \quad \hat{y}'_i = \frac{y'_i}{\sqrt{x'_i{}^2 + y'_i{}^2}}$$

- **Normalized second derivatives**: The second derivatives are computed by replacing x and y with $\hat{x}'$ and $\hat{y}'$ in the first part of formulae and normalized similarly.
- **Curvature**: Curvature at a point on a plane curve is defined as the inverse of the radius of the osculating circle. It is calculated as

$$\kappa_i = \frac{\hat{x}'.\hat{y}'' - \hat{x}''.\hat{y}'}{\left(\hat{x}'^2 + \hat{y}'^2\right)^{3/2}}$$

- **Aspect**: Aspect at a point characterizes the ratio of the height to the width of the bounding box containing points in the neighborhood. It is computed as in NPen++ [8]. It is given by

$$A(t) = \frac{2 \times \Delta y(t)}{\Delta x(t) + \Delta y(t)} - 1$$

where $\Delta x(t)$ and $\Delta y(t)$ are the width and the height of the bounding box containing the points in the neighborhood of the point under consideration. In all our experiments, we have used a neighborhood of length 2 i.e. two points to the left and two points to the right of the point along with the point itself.

- **Curliness**: Curliness at a point gives the deviation of the neighborhood points from the line joining the first and last points in the neighborhood. It is given by

$$C(t) = \frac{L}{\max(\Delta x, \Delta y)} - 2$$

where L is the sum of all the line segments along the trajectory in the neighborhood of the point [8].

- **Lineness**: It is the average squared distance between every point in the neighborhood and the line joining the first and last points of the neighborhood [8].

4.2. Frequency-domain features

To determine the frequency domain features, the character sequence is viewed as a complex function $f : t \rightarrow (x_t + iy_t)$, where t denotes time and x_t and y_t are the coordinates of the point at time t.

The frequency domain features were computed along the stroke using a sliding Hamming window. At each point, the window is centered and Discrete Cosine Transform (DCT) is evaluated on the windowed sequence. The real and imaginary parts of the lowest 4 coefficients excluding DC coefficient were added to the feature vector. The number of coefficients to be considered was determined empirically.

5. Classification

5.1. HMM Classifier

A Hidden Markov Model is a doubly stochastic model [9]. The underlying stochastic process corresponds to state transitions that are hidden, but the state changes are observed through another set of stochastic processes that produce the output symbols. The output symbol is said to be discrete if it is from a finite alphabet, and it is continuous if it has real-valued attributes. Accordingly, the model is called discrete or continuous HMM. In our experiment, continuous HMMs were used to model the Telugu symbols since the features are real-valued. The most commonly used HMM topology for both speech and handwriting is the left-to-right model, also known as the Bakis model [9]. It takes into account the temporal order of the signal.

An HMM state is said to generate feature vectors following a probabilistic distribution, usually a mixture of Gaussians. The number of Gaussians in the mixture and the number of states in the HMM were determined empirically. HMM training was done using the well-known Baum-Welch re-estimation procedure [9]. In our experiment a total of 141 HMMs corresponding to 141 symbol classes were trained. Given a test symbol, the probability associated with each one of the symbol-HMMs was computed and the symbol that has the maximum probability is declared was the recognition result. The probability associated with each symbol was computed using the HMM forward algorithm [9].

6. Performance Evaluation

The recognition system was trained and tested on the Telugu symbol data, collected following the procedure described in Section 2. The training set contained a total of 29,158 samples collected from 108 writers with approximately 200 samples per class. The test set contained approximately 70 samples per class amounting to 9,235 samples collected from a different set of 35 writers. The number of states per HMM and the number of Gaussians per mixture were determined empirically. The number of states per HMM was found to be 6. Table 1 summarizes the results obtained.

Table 1. Recognition performance of the system for various number of Gaussians per state with Time-domain features (TDF) alone, Frequency domain features (FDF) alone and their combination.

Gauss	Features	ACCURACY (%) Top-1	Top-2	Top-3	Top-4	Top-5
4	TDF	84.8	94.4	96.7	97.8	98.4
	FDF	88.8	95.7	97.1	97.7	98.0
	TDF+FDF	89.7	96.2	97.6	98.2	98.5
8	TDF	88.0	95.8	97.5	98.1	98.7
	FDF	90.0	96.2	97.5	98.0	98.2
	TDF+FDF	90.6	96.6	97.8	98.4	98.7
12	TDF	89.6	95.9	97.5	98.2	98.5
	FDF	90.3	96.3	97.5	97.9	98.2
	TDF+FDF	90.8	96.7	97.9	98.4	98.7
16	TDF	88.2	95.7	97.2	98.0	98.4
	FDF	90.4	96.3	97.5	97.9	98.1
	TDF+FDF	**91.6**	97.0	98.0	98.4	98.7
20	TDF	88.7	96.0	97.6	98.2	98.6
	FDF	89.9	96.1	97.3	97.6	97.8
	TDF+FDF	91.3	96.9	97.9	98.4	98.7

The highest top-1 accuracy of 91.6% was obtained for a mixture of 16 Gaussians per state. The results clearly show that the recognition rate obtained by combining frequency and time domain features is better than with either of them alone. We have also observed that a large number of misclassifications were due to similar-looking symbols and poor writing. Some of the misclassified symbols are shown in Fig. 3.

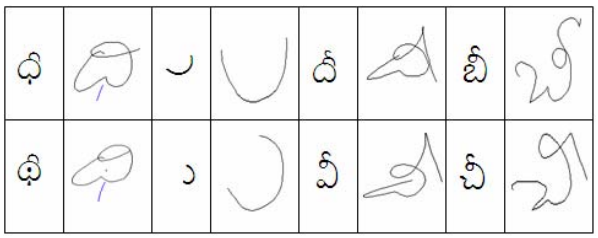

Figure 3. Some of the confused pairs

The performance of the system may be improved by employing discriminatory classifiers trained on most confusing pairs, in a second stage after HMM recognition. Manual analysis of the recognition results reveal that a few classification errors were due to stroke order variation. Such misclassifications were only restricted to a smaller number of classes, indicating that most of the symbols were free from stroke order variation. In order to address the issue of a symbol being written in different stroke orders, the symbol can be modeled using as many HMMs as there are stroke orders.

7. Conclusions

In this paper, we described an online handwriting recognition system for Telugu symbols based on Hidden Markov Models. A data collection procedure based on ACECAD Digimemo was discussed and its application to Telugu data collection was illustrated. The combination of time-domain and frequency-domain features was shown to yield better results than using either of them individually.

In the future, we would like to extend the system for recognition of complete syllabic units and words in Telugu. Even though the system has been tested only on Telugu data, the approach is completely data-driven and has no script-dependent features. Therefore the technique can also be extended to other Indic scripts. Another research direction for the future is to investigate features that would better discriminate the confusing pairs.

8. References

[1] M. Srinivas Rao, Gowrishankar, V.S. Chakravarthy, "Online Recognition of Handwritten Telugu Characters," *Proceedings of the International conference on Universal Knowledge*, 2002.

[2] Hariharan Swethalakshmi, Anitha Jayaraman, V. Srinivasa Chakravarthy and C. Chandra Sekhar, "Online Handwritten Character Recognition of Devanagiri and Telugu Characters using Support Vector Machines," *Proceedings.of the 10th International Workshop on Frontiers in Handwriting Recognition,*October 2006..

[3] F. Coulmas, "The Blackwell Encyclopedia of Writing Systems," Blackwell, Oxford, 1996.

[4] Mudit Agrawal, Ajay S Bhaskarabhatla, Sriganesh Madhvanath, "Data Collection for Handwriting Corpus Creation in Indic Scripts," *Proceedings of the International Conference on Speech and Language Technology and Oriental COCOSDA (ICSLT-COCOSDA 2004)*, New Delhi, India, November 17-19, 2004.

[5]ACECAD DigiMemo A502, http://www.acecad.com.tw/dma502.html

[6] UNIPEN format, http://unipen.nici.ru.nl/unipen.def

[7] M.Pastor, A. Toselli, and E.Vidal, "Writing Speed Normalization for On-Line Handwritten Text Recognition," *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR),* 2005.

[8] S.Jaeger, S. Manke, J. Reichert, A. Waibel, "Online handwriting recognition: the NPen++ recognizer," *International Journal on Document Analysis and Recognition,* March, 2001, vol.3 (3,) pp. 169-180

[9] Rabiner R. "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of IEEE*, 1989, 79(2). pp. 257-286.